

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

- 1 1. (Currently Amended) A computer-implemented system for detecting termination of
2 an application instance using locks, comprising:
3 a holding process configured to obtain a first exclusive lock on an object maintained
4 by the application instance
5 a waiting process configured to (a) request a second exclusive lock on the object
6 after the holding process has been granted the first exclusive lock on the
7 object, and (b) return a result signal, to a monitor process, upon at least one
8 of acquiring the second exclusive lock and ceasing to be blocked; and
9 the monitor process configured to process the result signal to determine whether the
10 application instance has terminated.
- 1 2. (Previously Presented) A system according to Claim 1, further comprising:
2 the monitor process determining whether the application instance terminated based,
3 at least in part, on whether the monitor process receives a standard error or a
4 non-standard error from the waiting process.
- 1 3. (Previously Presented) A system according to Claim 1, wherein the holding process
2 resides at the same node as the application instance, and where the waiting process
3 does not reside at the same node as the application instance.
- 1 4. (Previously Presented) A system according to Claim 3, further comprising:
2 a validation module configured to (a) check for termination of the monitored
3 application and (b) signal termination of the monitored application to a cluster
4 service.

- 1 5. (Previously Presented) A system according to Claim 3, further comprising:
2 a validation module configured to (a) check for termination of the monitored
3 application and (b) restart the holding process and the waiting process.
- 1 6. (Previously Presented) A system according to Claim 1, wherein the application
2 instance is a database service instance.
- 1 7. (Currently Amended) A computer-implemented method for detecting termination of
2 an application instance using locks, comprising:
3 starting a holding process configured to perform the steps of
4 (a) acquiring a first exclusive lock on an object maintained by the application
5 instance, and
6 (b) returning a ready signal, to a monitor process, upon successfully
7 acquiring the first exclusive lock; and
8 in response to receiving the ready signal, starting a waiting process configured to
9 perform the steps of:
10 (a) connecting to the application instance,
11 (b) requesting a second exclusive lock on the object maintained by
12 application instance, and
13 (c) returning, to the monitor process, a result signal upon at least one of
14 acquiring the second exclusive lock and ceasing to be blocked; and
15 processing the result signal, at the monitor process, to determine whether the
16 application instance has terminated.
- 1 8. (Previously Presented) A method according to Claim 7, further comprising:

2 determining whether the application instance terminated based, at least in part, on
3 whether the monitor process receives a standard error or non-standard error
4 from the waiting process.

1 9. (Previously Presented) A method according to Claim 7, wherein the holding process
2 resides at the same node as the application instance, and wherein the waiting process
3 does not reside at the same node as the application instance.

1 10. (Previously Presented) A method according to Claim 9, further comprising the steps
2 of:
3 checking for termination of the application instance; and
4 signaling termination of the application instance to a cluster service.

1 11. (Previously Presented) A method according to Claim 9, further comprising the steps
2 of:
3 checking for termination of the monitored application; and
4 restarting the holding process and the waiting process.

1 12. (Previously Presented) A method according to Claim 7, wherein the application
2 instance is a database server instance.

1 13. (Previously Presented) A machine-readable medium carrying one or more
2 sequences of instructions, which when executed by one or more processors, causes
3 the one or more processors to perform the steps of method recited in Claim 7.

1 14-26. (Cancelled).

1 27. (New) A machine-readable medium carrying one or more sequences of instructions,
2 which when executed by one or more processors, causes the one or more processors
3 to perform the steps of method recited in Claim 8.

1 28. (New) A machine-readable medium carrying one or more sequences of instructions,
2 which when executed by one or more processors, causes the one or more processors
3 to perform the steps of method recited in Claim 9.

1 29. (New) A machine-readable medium carrying one or more sequences of instructions,
2 which when executed by one or more processors, causes the one or more processors
3 to perform the steps of method recited in Claim 10.

1 30. (New) A machine-readable medium carrying one or more sequences of instructions,
2 which when executed by one or more processors, causes the one or more processors
3 to perform the steps of method recited in Claim 11.

1 31. (New) A machine-readable medium carrying one or more sequences of instructions,
2 which when executed by one or more processors, causes the one or more processors
3 to perform the steps of method recited in Claim 12.